ZURU™

ZURU TECH ACADEMY 2022
Third Edition March 1st 2022

MODULES

C++ Fundamentals:

1. syntax and introduction
2. memory managing and general differences with Java or C#
3. differences with C language
4. strings (char sequences), C Array and intro on Standard Library
5. data type (for academists who know only weakly typed languages)
6. pointers and references
7. uses of "&" and "*" (&x, x&, x*, *x)
8. lvalues and rvalues
9. exceptions (differences with other languages)
10. preprocessor
11. base compilaer (compiling from CLI + flag overview)

INTERMEDIATE C++:

1. Compiler, Linker and Libraries
2. class and struct (differences with other languages)
3. incapsulations (access specifiers) (important for python or javascript users)
4. const correctness
5. “Static” keyword
6. overloading operators and functions
7. Interface
8. polymorphism (with real case studies)
9. component over inheritance (multiple inheritance, diamond problem, etc)
10. Template
11. rule of 0/3/5
12. Casting
13. STL(+ Unreal stl-like classes comparison)
14. multithread base

ZURU™

ADVANCED C++:

1. type deduction (auto, decltype)
2. enum class
3. Nullptr
4. uniform initialization
5. no except
6. rvalue reference and move semantic
7. Lambda
8. algorithms and containers
9. user defined literals
10. miscellaneous on c++11 (range-based loops, differences on (">>" vs "> >"), integers types, new literals etc)
11. default delete
12. attributes ([[foo]])
13. Multithreading
14. smart pointers
15. perfect forwarding

SW GENERAL:

1. Coding style differences and Zuru Tech Coding Style
2. Design Patterns and Antipatterns
3. test infrastructure and basic concepts (unit, integration and functional testing + QA)
4. Best Practices

UE & PROCEDURAL MODELING:

1. Points, triangolations & primitives generation
2. UE ProceduralMeshComponent
3. Object generation
4. Procedural UV

UE:

1. quick start
2. blueprint system
3. C++ dialect and build system
4. Spec test framework
5. materials (master material and material instances)
6. UI/UMG

7. shaders (via nodes, via code (custom node), ray marching fundamentals)

GIT (all CLI):

1. Fundamentals (status, add, commit, checkout)
2. Branches and remotes
3. rebase (normal and interactive)
4. GIT LFS

CMAKE:

1. Fundamental concept of a build system, fundamentals of Cmake
2. Configure and build a cmake project

Professor:

Andrea Ruzzenenti - Senior Software Engineer & Teacher
EMail: andrea.ruz@zuru.tech

Company reference:

Silvia Pistoni - HR Recruiter
EMail silvia.p@zuru.tech Mobile: +39 348 358 2839

Tech Lead & Supervisor

Paolo Galeone - Machine Learning & Computer Vision Researcher
EMail: paolo@zuru.tech

ZURU™

ZURU TECH ACADEMY 2022
Terza Edizione 1 Marzo 2022

MODULI

BASE C++:

1. introduzione e sintassi
2. memory managing e differenze generali con Java o C#
3. Differenze generali con C
4. strings (char sequences), C Array e intro su Standard Library
5. data type (per i partecipanti che conoscono solo weakly typed languages)
6. pointers e reference
7. usi di "&" e "*" (&x, x&, x*, *x)
8. lvalues e rvalues
9. exceptions (difference con altri linguaggi)
10. Preprocessore
11. Compilatore base (compilare da CLI + flag overview)

INTERMEDIATE C++:

1. Compilatore, Linker e Librerie
2. class e struct (differenze con altri linguaggi)
3. incapsulamento (access specifiers) (importante per utenti python o javascript)
4. const correctness
5. “Static” keyword
6. overloading operatori e funzioni
7. Interfaccia
8. polimorfismo (con reali casi d'uso)
9. component over inheritance (multiple inheritance, diamond problem, etc)
10. Template
11. rule of 0/3/5
12. Casting
13. STL(+ Unreal stl-like classes comparison)
14. multithread base

ZURU™

ADVANCED C++:

1. type deduction (auto, decltype)
2. enum class
3. Nullptr
4. uniform inizialization
5. no except
6. rvalue reference e move semantic
7. Lambda
8. Algoritmi e container
9. user defined literals
10. varie c++11 (range-based loops, differenze di sintassi (">>" vs "> >"), tipi di interi, nuovi literals etc)
11. default delete
12. attributi ([[foo]])
13. Multithreading
14. smart pointers
15. perfect forwarding

SW GENERAL:

1. differenze tra vari coding style, visione del Coding style Zurutech
2. Design Pattern e Antipattern
3. test infrastructure and basic concepts (unit, integration and functional testing + QA)
4. Best Practices (generali)

UE & PROCEDURAL MODELING:

1. punti, triangolazioni & generazione di primitive
2. UE ProceduralMeshComponent
3. Generazione Oggetti
4. Procedural UV

UE:

1. quick start
2. blueprint system
3. dialetto C++ e build system
4. Spec test framework
5. materiali (master material e material instances)
6. UI/UMG
7. shaders (via nodi, via codice (custom node), basi di ray marching)

ZURU™

GIT (all CLI):

1. base (status, add, commit, checkout)
2. Branches and remotes
3. rebase (normale e interattivo)
4. GIT LFS

CMAKE:

1. concetti base di cosa sia un build system, concetti base di cmake
2. configurare e buildare un progetto cmake

Docente del corso:

Andrea Ruzzenenti - Senior Software Engineer & Teacher
Mail: andrea.ruz@zuru.tech

Referente organizzativo:

Silvia Pistoni - HR Recruiter
Mail silvia.p@zuru.tech Mobile: +39 348 358 2839

Tech Lead & Supervisor

Paolo Galeone - Machine Learning & Computer Vision Researcher
Mail: paolo@zuru.tech